ARI Research Note 97-14

# Skilled Use of Computer Software: Implications for Training and Design


**Judith Olson**
University of Michigan

**Peter Polson**
University of Colorado


**Research and Advanced Concepts Office**
**Michael Drillings, Chief**

**June 1997**

DTIC QUALITY INSPECTED 2

19970825 129

ARMY RESEARCH INSTITUTE

**United States Army**
**Research Institute for the Behavioral and Social Sciences**

Approved for public release; distribution is unlimited.

# U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

## A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

**EDGAR M. JOHNSON**
**Director**

---

Research accomplished under contract
for the Department of the Army

University of Michigan

Technical review by

Michael Drillings

**NOTICES**

**DISTRIBUTION**: This report has been cleared for release to the Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or the National Technical Information Service (NTIS).

**FINAL DISPOSITION**: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

**NOTE**: The views, opinions, and findings in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other authorized documents.

# REPORT DOCUMENTATION PAGE

| | | |
|---|---|---|
| 1. REPORT DATE<br>1997, June | 2. REPORT TYPE<br>Final | 3. DATES COVERED (from. . . to)<br>December 1988-December 1993 |

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br><br>Skilled Use of Computer Software: Implications for Training and Design | 5a. CONTRACT OR GRANT NUMBER<br>MDA903-89-K-0025<br><br>5b. PROGRAM ELEMENT NUMBER<br>0601102A |
| 6. AUTHOR(S)<br><br>Judith Olson (University of Michigan) and Peter Polson (University of Colorado) | 5c. PROJECT NUMBER<br>B74F<br><br>5d. TASK NUMBER<br>1901<br><br>5e. WORK UNIT NUMBER<br>C19 |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>University of Michigan<br>DRDA<br>13181SR<br>Ann Arbor, MI 48109-1248 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>U.S. Army Research Institute for the Behavioral and Social Sciences<br>ATTN: PERI-BR<br>5001 Eisenhower Avenue<br>Alexandria, VA 22333-5600 | 10. MONITOR ACRONYM<br>ARI<br><br>11. MONITOR REPORT NUMBER<br>Research Note 97-14 |

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

COR: Michael Drillings

14. ABSTRACT *(Maximum 200 words)*:

This research had two goals: to work toward developing a comprehensive cognitive theory of human-computer interaction, both learning and performance, and to develop methods for designers to apply this knowledge to the design of new software and its training. The researchers compare the actual moment-by-moment activity of users of software with predictions from a model, leading to extensions of the model and additional rounds of empirical testing. At the end of the project, more was understood about the development of expertise in people who use software, focusing in particular on how people build on what they know already when they encounter a new application. In the second methodology, designers follow concrete, well-specified steps that guide them through a set of analyses--first of the users' tasks, and then highlighting aspects of the target system that the model tells us may prove difficult for the user to learn or perform. The researchers conclude with an attempt to develop methods for designers to help them assess how long it will take to learn the software, based both on the complexity of the software and how much the learner already knows.

15. SUBJECT TERMS

Human-Computer Interaction    Computer Training and Design    Cognitive Models

| SECURITY CLASSIFICATION OF | | | 19. LIMITATION OF ABSTRACT | 20. NUMBER OF PAGES | 21. RESPONSIBLE PERSON (Name and Telephone Number) |
|---|---|---|---|---|---|
| 16. REPORT<br>Unclassified | 17. ABSTRACT<br>Unclassified | 18. THIS PAGE<br>Unclassified | Unlimited | 17 | Dr. Michael Drillings<br>(703) 617-8641 |

SKILLED USE OF COMPUTER SOFTWARE: IMPLICATIONS FOR TRAINING AND DESIGN

## CONTENTS



## LIST OF TABLES

# FINAL REPORT

Grant Period: December, 1988 through December 1993

# Skilled Use of Computer Software: Implications for Training and Design

MDA903-89-K-0025

CO-PRINCIPAL INVESTIGATORS:
Judith Olson, The University of Michigan
and
Peter Polson, The University of Colorado

## OVERVIEW OF PROJECT

**Problem.** Many functions in the modern military require the use of powerful and complex computer software running on increasingly sophisticated computer systems. However, the productivity gains and costs savings that were promised by developers are rarely achieved because training costs are high and because personnel rarely master these complex systems.

**Solution.** Design of software that is more effective as well as easier to learn requires a) a better understanding of the cognitive capabilities of users, and b) methods for designers to apply this knowledge in the process of designing software and its training. The theory and application methods that are the focus of this research project will provide developers with tools necessary to build highly functional systems and train users of these systems in an effective, efficient manner.

**Goals** This project had two goals: to work towards developing a comprehensive cognitive theory of human-computer interaction, both learning and performance, and to develop methods for designers to apply this knowledge to the design of new software and its training.

The first goal is to understand the cognitive capabilities of the user. This understanding comes from an interplay of **empirical work** and the development of an increasingly **comprehensive cognitive theory** of the acquisition and execution of the skills required to use complex software tools. In our empirical work, we compare the actual moment-by-moment activity of users of software with predictions from a model, leading to extensions of the model and additional rounds of empirical testing. The models are based on GOMS and Cognitive Complexity Theory (CCT), profiting from ten cumulative years of cognitive modeling efforts in our laboratories. Our near-term modeling extensions have added to GOMS with more of the perceptual, memory, planning, and motor movement components of behavior, allowing us to better predict users' behavior in their work context. At the end of the project, we moved toward understanding the development of expertise in people who use software, focusing in particular on how people build on what they know already when they encounter a new application.

A second goal is to **develop methods** that help the designer apply this understanding during the design process, methods that apply to both new applications and training materials. The design methodology follows the spirit of methods developed by Polson and Lewis and their colleagues (Lewis, Polson, Wharton and Rieman, 1990; Polson, Lewis, Rieman, and Wharton, 1992). In this methodology, designers follow concrete, well specified steps which guide them through a set of analyses-- first of the users' tasks, and then highlighting aspects of the target system that our model tells us may prove difficult for the user to learn or perform. We ended with an attempt to develop methods for designers to help them assess how long it will take to learn the software, based both on the complexity of the software and how much the learner already knows.

## EMPIRICAL WORK

The goal of the empirical work was to collect data on the aspects of the human information processor that were heretofore unexplored. It is the tradition in experimental/cognitive psychology to assess processes in isolation of each other, and with tasks that are specifically constructed to decide between two alternative models. Here, we needed to expore phenomena as they occur in concert with other cognitive processes, and in context that are similar to real world behavior. The intent is to build an empirical base of knowledge that would help us understand the behavior of people using computer applicatins and to turn this model to the practical use of designing user interfaces and training methods.

We explore seven such processes in our empirical work:

> The cognitive processes and timing involved in how people choose among various methods to accomplish goals,
>
> The associated cost of learning specialized methods in the light of the way people decide to use them in the future,
>
> The minute planning and execution details of hand motions involved with various types of mouse input devices and variously designed menus,
>
> The perceptual and working memory processes involved in seeing and understanding a visual display to answer a specific question,
>
> The working memory load from planning and executing complex commands,
>
> The ways in which these processes change as the user develops skill, and
>
> Components of learning systems that have various kinds of similarity to each other.

These studies join to flesh out the model of the human information processor, exploring in more detail that the work of Card, Moran, and Newell, the processes of perception, motor movement and cognition.

**How people choose among methods and the time it takes to make that choice.** *(Nilsen, Jong, Olson, and Polson, 1992)* We presented experts with situations for which there were more than one method to choose from. We constructed situations that varied along dimensions such that at some point one method was more time-efficient (determined using GOMS keystroke analyses). For example, we asked the experts to edit some labels and formulas in a variety of locations on an existing spreadsheet. These task could be done with several methods, some appropriate for very small changes or navigations, some for large. We calculated for each situation the "ideal method", the method that, according to GOMS keystroke analysis, was the shortest one to execute. We similarly calculated "choice cusps," situations in which several methods were equally efficient.

We collected large amounts of data from 6 "power users," looking to see if their choice of methods followed the predictions from the simple heuristic of choosing the most efficient method. As predicted, subjects switched methods near the point at which the methods were equally efficient. In most cases, however, the switch was made slightly earlier than we had anticipated; they chose the specialized method ("edge" in this case) more often than would be predicted by an efficiency analysis. In addition, the subjects were more variable than anticipated, both across subjects and within subjects. Subjects' base rates of choosing any particular method differed significantly . Also, individual subjects used different choicepoints in ostensibly equivalent tasks. When navigating *between* tasks, far more subjects choose the mindless "arrow key" method, whereas in specifying the range (which is formally equivalent to moving between tasks), they were more likely to choose the more efficient, but harder to remember "edge" command or the method that allows you to type in a cell name or coordinate .

Clearly, factors other than efficiency were influencing the choice. We have developed a model that describes the number of ways in which people could choose a method for the current situation (Nilsen, et al, 1992), and are now writing the full report of the empirical work.

**The extra cost of learning multiple methods.** *(Ashworth, 1992)* This study provides us with estimates of the training costs of offering multiple ways of performing a task in a system, and confirming the performance costs from the study of experts above. We already know that learning more methods requires that subjects learn more individual steps; these steps can be described in terms of CCT. However, we also know that the steps to be learned are very similar, suggesting there may be an additional cost in terms of associative interference.

In this study we taught new users either one or two methods for completing various common tasks, such as the navigation task illustrated in the expert study above. By assessing the learning time and the pauses immediately before the method is executed, we can better assess the overall learning and performance costs of offering multiple methods.

**The time to plan and execute various complex motor movements involved in various kinds of hierarchical menu designs.** *(Nilsen, 1991)* This series of studies extends the ideas of GOMS into the realm of menu selection with mice. Previous detailed analyses of the motor movements involved in the use of mice suggests inadequacies in Fitt's law. In particular, two new kinds of menus, common to recent commercial software, are more difficult to navigate than a straight application of Fitt's law would predict: "walking menus" and "click open menus." These studies extend what is known about motor movements to cover these two menu types, and at the same time build a model that may help us to assess the adequacy of a number of new types of menus as yet unexplored.

In the first experiment, subjects were presented with a goal as to what they were to select in one of two kinds of menus:

> a) the "walking menu" required the subject to begin the traversal of the menu with the mouse button down and to hold it down while a complex path was traveled. The path included arriving at the right-hand portion of a menu item, at which time the second level menu for that item appeared. The subject then passed through a "gate" at the right hand side of that item and then traversed down the second menu until the cursor was on top of the correct item. At that point, and only at that point, could the mouse button be released.
>
> b) the "click-open" menu requires many more button press-releases, but a less complicated path. In these menus, the subject opens the first level menu with a click (press-release), then moved down to the desired first-level menu item, clicked again to open the second-level menu, then moved to the final desired item and clicked again.

The first study, subjects showed that although the "click-open" menu required more discrete acts, it was faster by 10%. This study also found that when the subjects knew in advance which position the target was going to be in, there was a large effect on the total time. When users know the position of the item, the timing is well modeled by Fitt's law. Without advance information (e.g., when someone knows the menu name but not where it will appear in the menu), the selection time is dominated by a self-terminating top-down visual search. The interplay of these components is initially thought to be modeled using Critical Path Analysis, which was introduced to this kind of phenomena by John (1988).

A second study explores the causes of the above differences by using only single level menus. Because it found the walking menu faster in total by 6%, similar to the 10% advantage above, we conclude that the difference is due to holding the mouse button down while traversing, not to the fact that on two-level menus, one has to traverse a very narrow path to open the second level menu.

A third and fourth studies were conducted to try to pin down the mechanism that caused this motor disadvantage in holding the mouse button down. It turns out that it is best explained by additional muscle contraction, not anything more central (like remembering to release the downed mouse button) or physical (like friction from the mousepad).

**The time and effort involved in understanding information on a visual display.** ***(Lohse, 1990 and 1991)*** We have constructed a model of the processes that someone goes through to understand a visual display and to answer questions about it. The model of this process includes

- **parsing and encoding the question** to be asked or the goal to be accomplished
- **retrieving a method** by which this display can be searched for relevant information,
- **searching the display** to read relevant values and comparing them, if necessary, to answer the question asked.
- at each point where the presentation is difficult to discern, **discriminating** the lines, bars, or table-values to determine the accurate value.
- building an answer to the posed question, keeping partial results in a **working memory**

The model includes values for the time it takes the eye to move from one location to another, the time it takes to register a corresponding symbol to a label in a "legend," the time to discriminate one symbol from another, whether they be color or shapes, the time enhancement in eye movements for being able to follow grid lines or meaningful white spaces, etc. Many of the process descriptions and the timing parameters were gleaned from the cognitive/perceptual psychology literature, along with well-known principles of perceptual understanding like Gestalt principles, etc. Included also is a working-memory stack that holds partial answers until the full form answer can be formulated.

The model has been subjected to rigorous empirical verification. Reaction times to yes/no questions of three types (point reading, comparisons, trends) were collected from 28 subjects. Each subject provided 576 observations representing eight replications over a variety of situations. The overall mean time predicted is within 262 msec of what the subjects did, within a keystroke time. Since the model doesn't include that final keystroke, we take this as a sign of very good overall fit. Regression analysis predicts about 58 percent of the variation. The largest predictive component is discrimination time, explaining 36 percent of the variation by itself. Cognitive components explain the next largest proportion of the variance; scanning time accounts for very little. These results have practical implications. To significantly improve the readability of graphs, designers should spend more effort on making sure the target points are discriminable when they appear within a fixation.

A second, more far reaching study has been run. In this study we pursued the simple practical question: So what? This study translates the time or effort saved in perceiving and understanding a graph into implications for how the user will make critical decisions from the graph. The underlying premise is that the more effort the user has to expend in understanding the graph, the less left over for deeper considerations about the decision itself. For this study we collected the performance of MBA students in making a decision about how much money different sectors of a company will receive for marketing. The presentations (graphs, tables, with and without color and grid lines) that give feedback from one decision to the next are designed under guidance from the model above to be of various levels of difficulty. The quality of the decision clearly drops with complexity, and the subtle details of the drop follow closely the predictions of the model of perceptual effort.

**The Cognitive Load involved in planning and executing complicated sequences of actions.** ***(Jong, 1991)*** The fifth set of studies investigates the stage when the user translates the goal into a complicated action specific to the piece of software and then monitors the execution of that complicated action. Two specific general tasks are used to investigate this translation process, each in a different study: the specification of complex relationships such as in a spreadsheet formulas, and the specification of a complex command, such as setting a range of columns in a spreadsheet to a particular width.

We have investigated the processes involved when the way the users state the goal in their heads (e.g., in a sentence) mismatches the *order* in which items must be specified in the input language of the computer. There are two possible ways the user may handle this difficulty: translate the entire goal into the sequence to be generated before executing (in a spirit similar to the external-internal-task mapping of Moran, 1983), and translating "on the fly", generating the beginning of the sequence, then checking and re-planning as you go to make the sequence right. In both cases, the order-mismatch puts a burden on the intermediate memory, likely precipitating errors or requiring the user to generate adaptive tactics for scanning the environment and checking for errors.

In the first two studies, people were required to enter complex formulas (e.g., ((A+B)/C + D*F) in one of two ways. The first is the left-to-right (L-R) embedded parentheses notation familiar to current users of spreadsheet software such as Lotus 1-2-3. The second is a new editor that has formulas entered as triples, object-action-object (OAO), in which each such triple can serve as a higher-level object for other actions. The former puts great burden on the user to keep track of the open and closed parentheses. The second order allows the user to think of the natural units of the task and to "close" each before starting the next. For the subjects, an average formula took nearly half the time with the OAO editor, 32 sec vs. 62. Furthermore, there were many more extra keys pressed with the traditional LR editor (for backing up and inserting earlier missed parentheses), 16 times as many.

In the third study, the idea of "natural order" of entry was extended to include command sequences other than formulas. Here, the complex menu of Lotus 1-2-3 was analyzed for the order of the constituents in the command: either ***action on an object*** or *for this* ***object****, perform this* ***action***. Two groups of subjects were taught two organizations of the same menu items as found in Lotus 1-2-3, the traditional order (which is actually mixed in its grammar) and a revised menu that was consistently *object-action.* The 30 Ss with the revised menu performed the tasks in 1/3 the time it took 30 Ss with the traditional menu. Furthermore, when give the task goal in the order that fit the menu order (in words other than the words used in the menus), Ss were even faster yet.

**The Development of Software Skill.** *(Nilsen, Jong, Olson, Biolsi, Rueter, and Mutter, 1993)* A longitudinal study of people learning spreadsheet software was run a number of years ago, but the analysis of the data's correspondence with an extended GOMS-keystroke model was completed over the past year. Thirty five MBA students were tested four times in two years to assess their performance on four isomorphic spreadsheets. We found that as experience increases, the simple motor components of performance did not change whereas the mental components (retrieval and planning) were much improved. Furthermore, the mental components of more frequently performed tasks appear to improve more rapidly, confirming and refining well known laws of general learning.

It is this study that motivated a number of ideas in our modeling of how people choose among methods and the more detailed data collection efforts on learning and performance that appear in the other studies listed above.

**Learning of New Skills** *(Nilsen, et al, 1993, Polson and Olson, 1992)* First, we learned from a careful examination of a two-year longitudinal study (Nilsen, et al, 1993) that the progression of skill was systematic. Detailed analysis of the moment by moment timing of performance and the kinds of errors people exhibited showed that a number of different forms of learning and adaptation were going on. There was evidence for simple speedup of some subtask skills, mainly in more rapid access to the mental representations of the components. There was also evidence for chunking, where a series of actions were retrieved en masse and executed in a rapid series, very different from early more deliberate step-by-step performance. Third, there was a clear progression of people learning more--learning several ways to perform a particular subtask and also the conditions under which each would produce faster performance.

In a second study, we examined some ways in which to speed the learning of a new skill. The literature shows that if you tell a person how the device they are learning to operate works, it does not necessarily speed the performance of straightforward tasks, but does help them adapt to new tasks. (Kieras and Bovair, 1984, 1986) However, the addition of a description of a well-known system that works in a similar fashion (an analogous system) did not add any additional power to the new learning. We would have expected an increase in both speed of learning and accuracy of later retrieval because of transfer the known system analog was knowledge that had been already learned and merely needed reminding or associating to the new situation.

In an attempt to understand this result better, we ran a replicating study with a system that was both more complicated and, in one condition, taught with reference to a real world analogous that was as complex as the system to be learned. It was thought that the additional complexity would show the power of the analogous system. Learning materials were constructed that were comparable in every way (both words and diagrams) except for the inclusion of the labeling of the device components with the analog terms. However, once again, learning from the materials with information about how the device works was better than rote learning for transfer to new problems, but the extra explanatory prose, explaining that this device was like the analog system, did not provide any additional help.

## THEORY BUILDING

All of our modeling efforts revolve around the philosophy that one understands design issues in human-computer interaction by understanding how the user perceives, retrieves, and enacts within a system to get a task done. The steps that one goes through is shown in the diagram in Figure 1. below.

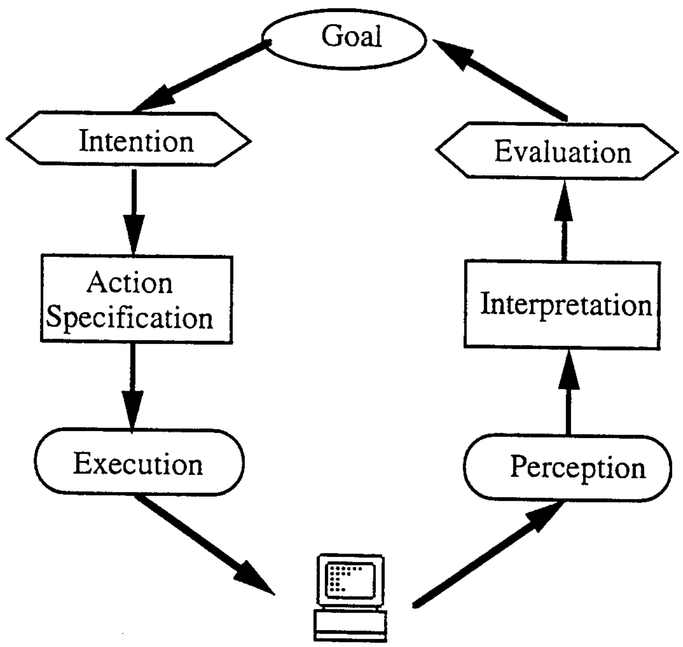


Figure 1. Norman's Action Theory (Adapted from Norman, 1986). The stages of processing of information in human computer interaction.

The user of a system is thought to have a goal in mind, and some knowledge about the characteristics of the system currently being used. The user retrieves a method that is appropriate to the particulars of the situation, loading the motor actions into a buffer, ready for execution. Having executed these actions, the user perceives the changes in the system. These changes are interpreted and matched with what was expected. The new situation is assessed, and the process cycles again.

The work in this project is currently evolving five models, each of which highlights several of the stages of this process, even though each has to account for all the stages in some fashion in order to predict behavior. For example, the model of **choice of methods** focuses on the mapping of information from the current situation and the goal to the retrieval of a method from memory; it makes simple assumptions about the motor movements involved and the perception of consequences of those actions. The **learning** model focuses on the changes that are anticipated when the user grows in experience, both initial learning and later performance with a system. The model on the planning and execution of **motor movements** that are involved when one chooses items from menus with a mouse includes simple forms of the stages of perception and goal formulation, but focuses in detail on the interaction of planning of movements and their sometimes difficult execution. The project on understanding **perceptual displays** focuses in detail on both the eye movements that drive the collection of information in a display, and the amount of intense computation that is or is not necessary when one tries to decipher particular information from a "busy" part of the display. It includes a simple action sequence, a simple yes/no motor response. And the model of the effort involved in **keeping track** of what has and has not been accomplished in a long, complex ongoing task focuses on the multi-cycle process, involving primarily perception and goal satisfaction, highlighting the use of a working memory for keeping track of where one is in a task.

Table 1. An indication of how the five models focus their modeling effort on the various stages of human computer interaction.

| | Method Choice | Learning Methods | Perceive Displays | Motor Movement | Cognitive Load |
|---|---|---|---|---|---|
| Perception | | | X | | x |
| Interpretation | | | X | | x |
| Evaluation | | | X | | x |
| Goal | X | X | X | | X |
| Intention | X | X | | X | X |
| Action Specification | X | X | | X | X |
| Execution | | | | X | |

Table 1 shows the seven components and the particular focus each model has chosen. The goal of the current work is to flesh out each of the individual models to account for a set of modeled phenomena in sufficient detail to be acceptable to both the psychological community and to have relevance in practical design considerations. Eventually, we may blend these models into a single comprehensive model of human-computer interaction, where some processes have more or less importance in the design of a particular feature of an interface than others, but none are excluded from consideration.

These component models are reviewed and integrated in a paper (Olson and Olson, 1990) that builds on the theoretical base begun by Card, Moran, and Newell, 1983), significantly fleshing out the perceptual, motor, and cognitive components.

## IMPLICATIONS FOR TRAINING AND DESIGN

Our attention has focused on helping designers on two important design issues: a) whether to invest the time to develop a new feature (a short-cut method or specialized macro capability like wordprocessor "styles") and b) how to assess how long a new application will take to learn.

The first of the methods (Nilsen et al, 1992) had at its core a cost/benefit analysis -- both for the end user ("will I choose to use this method which will give me this performance benefit given the cost of learning it") and for the designer ("should I design this method, given its cost in time and resources, if it has this calculated benefit to the user which implies a potential benefit in sales or corporate productivity").

The second method to help estimate how long a piece of software will take to learn (Polson and Olson, 1992). The answer to this, of course, is very complex. It clearly depends on how much new there is to learn, and how it is taught (how much the materials evoke existing knowledge). One could calculate how much new there is to learn if one could assess both what the learner knew already and exactly what had to be learned. This assessment, if it can be done precisely at all, would take an enormous amount of time--time much greater than can be invested by an everyday designer.

We constructed a method for this situation that has a number of heuristics for estimating these components. It has two parts to it: one to estimate the complexity of the skill to be learned; and one to assess the knowledge level of the learner. The assessment of the complexity of the skill to be learned is done by case-based reasoning. The designer finds an application that is judged similar in complexity (we give a set of examples in five classes). The designer then assesses the experience level of the learner, again in five classes. The method then has order-of-magnitude estimates of time to learn for each skill class. The designer does, essentially, a table look-up on this time and then adjusts downward for every grade of skill the learner has.

This method is appealing because of its potential simplicity. Although it is based on sound results from theory in cognitive psychology, it combines a lot of assumptions. It requires testing. Some of the proposed work for this year attempts to test the key components of this method.

## PUBLICATIONS


Ashworth, C. A. (1992). Skill as the fit between performer resources and task demands. In *Proceeding of the Fourteenth Annual Meeting of the Cognitive Science Society* . Hillsdale, N.J.: Lawrence Erlbaum Associates, Publishers. 444-449.

Jong, H. (1991) *The Subgoal Structure as a Cognitive Control Mechanism in a Human Computer Interaction Framework.* Doctoral Dissertation, University of Michigan.

Kitajima, M., and Polson, P. G. (1992) A computational model of skilled use of graphical user interfaces. *CHI '92 Conference Proceedings: Human factors in computing systems,* ACM, New York, pp. 241-249.

Lee, A.Y., Foltz, P.W., and Polson, P.G. (1992) Memory of Task-Action Mappings: Mnemonics, Regularity, and Consistency. ICS Technical Report No. 92-11, Boulder, CO.

Lewis, C., Polson, P., Rieman, J. Wharton, C., and Widle, N. (1992) Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. Tutorial Notes, *CHI '92 Conference on Human factors in computing systems,* ACM,

Lohse, G. (1991) A cognitive model for the perception and understanding of graphs. *Proceedings of the Conference on Human Factors in Computing Systems, CHI'91.*

Lohse, G. (1991) *A Cognitive Model for Understanding Graphical Perception.* Doctoral Dissertation, University of Michigan.

Nilsen, E. (1991) *Perceptual-Motor Control in Human Computer Interaction.* Doctoral Dissertation, University of Michigan.

Nilsen, E., Jong, H. S., Olson, J. S., and Polson, P. G. (1992) Method engineering: From data to model to practice. to appear in *CHI'92, Human Factors in Computing Systems.*

Nilsen, E., Jong, H. S. , Olson, J. S., Biolsi, K., Rueter, H, and Mutter, S. (1993) The growth of software skill: A longitudinal look at learning and performance.*CHI'92, Human Factors in Computing Systems.*

Olson, J. S. and Moran, T. (1995) Mapping the method muddle: Guidance in using methods for user interface design. In C. Lewis, Tim McKay, P. Polson, and M. Rudisill (Eds.) *Human-computer interface design: Success Cases, emerging methods, and real world contexts.* New York: Morgan Kaufman.

Olson, J. S., and Olson, G. M. (1990) The growth of cognitive modeling in human-computer interaction since GOMS. *Human Computer Interaction*, 5, 221-265.

Polson, P.G., Lewis, C., Rieman, J., and Wharton, C. (1992). Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Studies*, 36, 741-773.

Polson, P.G., Rieman, J., Wharton, C. and Olson, J.S. (1992) Usability Inspection Methods: Rationale and Examples. English *Proceedings of the SICE 8th Symposium on Human Interface. SICE*, Kawasaki, Japan. pp. 377-384.

Polson, P.G. and Olson, J.S. (1992) An Approximate Method for Estimating Total Learning Time: An Example of a Usability Inspection Method Derived from Cost/Benefit Considerations. A Paper presented at *Human-Computer Interaction Consortium* Annual Winter Conference, Pittsburgh, PA, January 28, 1992

APPENDIX A. PEOPLE INVOLVED IN THE PROJECT

**Principle Researchers**

Judith Olson,
co-principal investigator, University of Michigan

Peter Polson,
co-principal investigator, University of Colorado

**Graduate Students**

Erik Nilsen
graduate research assistant, University of Michigan

HeeSen Jong
graduate research assistant, University of Michigan

Jerry Lohse
graduate research assistant, University of Michigan

Catherine Ashworth,
graduate research assistant, University of Colorado

Mary Lassaline
graduate research assistant, University of Michigan

Robert Fein
graduate research assistant, University of Michigan

Andrea Galagher
graduate research assistant, University of Michigan

Susan McDaniel
graduate research assistant, University of Michigan

**Supporting Staff**

Henry Rueter
research associate, University of Michigan

Sue Schuon
administrative assistant, University of Michigan

Stacey Donohue
technical supervisor for the laboratory, University of Michigan

Lola McGuffin
technical supervisor for the laboratory, University of Michigan

Mark Carter
research assistant, University of Michigan

Jim Barbor
programmer, University of Colorado

Gwen Murphy
secretary, University of Colorado